Wikipedia Search Engine using various Embedding Techniques

Abhishek Nigam
abhishek22003@iiitd.ac.in
MT22003

Dhananjay Bagul
dhananjay22028@iiitd.ac.in
MT22028

Rohit Kesarwani
rohit22059@iiitd.ac.in
MT22059

Sushil Kumar
sushil22077@iiitd.ac.in
MT22077

Shambhavi Sharma
shambhavi22066@iiitd.ac.in
MT22066

Wilson Radadia
wilson22091@iiitd.ac.in
MT22091

Abstract—**Project aims to build a Wikipedia search engine to provide the relevant articles based on the user search query. The search engine will use various embedding techniques to generate embeddings for the input query and for the Wikipedia articles, which will be used to compute the relevance of the articles to the query. The baseline model aims to build a search engine to retrieve the most relevant Wikipedia articles based on a search query. The search engine uses two techniques: TF-IDF and Jaccard Coefficient. The updated project aims to build a Wikipedia search engine using the BERT model. The Performance of the search engine will be evaluated based on the given metrics - precision, recall, F1 score, MAP(Mean Average Precision), MRR(Mean Reciprocal Rank) and accuracy of the retrieved articles.These metrics will be used to assess the search engine's ability to retrieve the most relevant articles based on the user's query. Finally, this project seeks to create a more efficient and effective way for users to search for relevant information on Wikipedia, leveraging the power of BERT and advanced metrics to provide a more accurate and personalized search experience.**

Keywords—Wikipedia search engine, BERT, Embeddings, Relevance, Precision, Recall, F1 score, MAP, MRR, Accuracy.

I. Introduction

Projects like the BERT-based Wikipedia search engine with user feedback are significant to design because almost every user goes through a pervasive problem while searching for information online. This can happen due to many reasons like ambiguity, lack of relevant information, etc. By providing users feedback for the relevance of the results, we can create a more effective and interactive search experience on Wikipedia which makes it more useful for the users. Natural language processing models like BERT can help improve search results accuracy by going through the hints given by the language and generating more accurate embeddings for text.

The project is using the BERT model because it is a contextual model, which means that it can understand the meaning of words while that is not the case for TF-IDF as it is based on word frequency. Also TF-IDF is a fixed technique which does not improve over time while BERT can improve its performance over time.

II. Literature Review

"On Improving Wikipedia Search using Article Quality" by Meiqun Hu et al.[1] which proposed a framework that re-ranks search results based on article quality based on two development quality measurement models, namely Basic and PeerReview, based on co-authoring data gathered from articles' edit history.

The authors begin by discussing the difficulties associated with Wikipedia search, such as the presence of low-quality articles and a lack of user feedback. They then propose a novel approach to ranking search results that uses article quality as a relevance criterion. The proposed method is divided into two stages: the first is concerned with identifying high-quality articles, while the second is concerned with ranking the search results based on their relevance and article quality. Article quality is determined by a number of factors, including article length, edit frequency, and user ratings. The authors evaluate their approach using a large dataset of Wikipedia articles and user queries. The results show that incorporating article quality information into the search process improves the effectiveness of Wikipedia search, particularly in retrieving high-quality articles.

The paper makes an important contribution to the field of information retrieval by demonstrating the usefulness of incorporating article quality information into the search process. The proposed method could be applied to other large-scale collaborative knowledge bases like Wikidata and DBpedia. However, the paper does not provide a detailed analysis of the proposed approach's limitations or generalizability to other contexts. Overall, the paper offers useful insights into how to improve the effectiveness of Wikipedia search by taking article quality information into account.

"Leveraging BERT for Extractive Text Summarization on Lectures" by Derek Miller et al.[2] which proposes a mechanism for extractive summarization through the use of K-Means clustering algorithm and the BERT model for text embeddings to find the closest sentences.

The paper starts by describing the challenges that come with summarizing lecture transcripts, such as the length and complexity of the transcripts, as well as the need to capture important concepts and ideas. The authors then propose a BERT-based method for identifying and ranking important sentences in a transcript using contextualized word embeddings. The proposed approach is divided into three stages: the first involves pre-processing the transcript and encoding it with BERT; the second involves using a sentence scorer to identify important sentences in the transcript based on their contextualized embeddings; and the third stage involves selecting the top-ranked sentences to generate the summary. The authors compare their approach to several baseline methods using a dataset of lecture transcripts. The findings show that the proposed method outperforms the baseline methods in terms of ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores as well as human evaluations.

The paper contributes significantly to the field of text summarization by demonstrating the efficacy of using BERT for extractive summarization of lecture transcripts. The proposed method could be applied to other domains involving complex and technical text, such as scientific articles and legal documents. However, the paper does not provide a detailed analysis of the proposed approach's limitations or generalizability to other contexts. Overall, the paper provides useful information on how to leverage BERT for extractive text summarization.

"A survey on word embedding techniques and semantic similarity for paraphrase identification" by Divesh R. Kubal et al.[3] aims to provide an analysis of existing similarity metrics, statistical machine translation metrics and includes different word embedding techniques, stepwise derivation of its learning module.

The authors compare their approach to several baseline methods using a dataset of lecture transcripts. The findings show that the proposed method outperforms the baseline methods in terms of ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores as well as human evaluations. The paper contributes significantly to the field of text summarization by demonstrating the efficacy of using BERT for extractive summarization of lecture transcripts. The proposed method could be applied to other domains involving complex and technical text, such as scientific articles and legal documents. The authors then use a dataset of paraphrases to compare the performance of various similarity metrics and statistical machine translation metrics. The findings show that word embedding-based methods outperform traditional similarity metrics and statistical machine translation metrics in most cases.

The paper contributes significantly to the field of natural language processing by providing a thorough analysis of existing similarity metrics and statistical machine translation metrics for paraphrase identification, as well as an overview of various word embedding techniques and their learning modules. The paper is well-organized and gives a thorough overview of the various techniques and methods. Overall, the paper offers useful insights into how to use word embeddings for paraphrase identification.

“NLP based Intelligent News Search Engine using Information Extraction from e-Newspapers” Monisha Kanakaraj et al. [4] present a personalized news search engine that focuses on building a repository of news articles by applying efficient extraction of text information using TF-IDF from a web news page from varied e-news portals.

The paper first provides an overview of the challenges associated with news search, such as the need to accurately identify relevant articles and extract key information from them. The authors then propose a search engine that extracts information from e-newspapers and creates a database of relevant articles using NLP techniques. The proposed search engine has three major components: the first is crawling e-newspapers and extracting relevant information using NLP techniques; the second is storing the extracted information in a database and creating an index for efficient search; and the third is performing search queries using a user-friendly interface. The authors compare their search engine to several baseline methods using a dataset of news articles. According to the results, the proposed search engine outperforms the baseline methods in terms of accuracy and efficiency.

By demonstrating the effectiveness of using NLP techniques for information extraction and news search, the paper makes an important contribution to the field of information retrieval. The proposed search engine could be used in other domains with complex and technical text, such as scientific articles and legal documents. However, the paper does not provide a detailed analysis of the proposed approach's limitations or generalizability to other contexts. Overall, the paper offers useful insights into how to use NLP techniques for intelligent news search.

”Sarcasm Detection in Tweets with BERT and GloVe Embeddings” by Akshay Khatri, et al. [5] proposed a machine learning technique with BERT and GloVe embeddings to detect sarcasm in tweets.

The authors begin by discussing the difficulties in detecting sarcasm in text, particularly in short-form communication such as tweets. They then propose a methodology for capturing the semantic relationships between words and phrases in tweets and identifying sarcastic tweets using BERT and GloVe embeddings. A dataset of tweets labeled as sarcastic or non-sarcastic was used in the study. Several models, including logistic regression and neural network models, were trained and evaluated using BERT and GloVe embeddings. The results show that models with BERT embeddings outperformed those with GloVe embeddings, with an F1 score of 0.76. The authors also performed an error analysis to determine which types of tweets were difficult to classify. They discovered that tweets with negation, sarcasm within quotations, and sarcasm involving proper nouns were particularly difficult to correctly classify.

Overall, the paper provides useful insights into the efficacy of using BERT and GloVe embeddings for detecting sarcasm in tweets. The study shows that using these embeddings can improve the performance of sarcasm detection models significantly. The study, however, was limited to a single dataset of tweets, and more research is needed to assess the generalizability of the proposed methodology to other contexts.

"Using of Jaccard Coefficient for Keywords Similarity" by Suphakit Niwattanakul*, Jatsada Singthongchai, Ekkachai Naenudorn and Supachanun Wanapu et al. [6] paper proposes the significance of the Jaccard Coefficient for its different uses in different applications.

The paper provides an overview of different similarity measures. The different measures like Jaccard Coefficient, Euclidean distance, Cosine similarity with this the author explains about its application for measuring keyword similarity for different sets for particular documents. The authors then propose the outcome of the approach, that compares the performance of Jaccard Coefficient with other similarity measures. All the measures are good for keyword similarity but Jaccard Coefficient outperforms all other measures for keyword similarity. This makes Jaccard Coefficient a promising tool.

Overall, the paper provides the importance of the measurement and also demonstrates the efficiency of Jaccard coefficient. Also the author suggests that it can be a useful measure for the application of text categorization, document clustering and information retrieval. As it is more efficient in terms of accuracy and speed which makes it more useful.

”PASSAGE RE-RANKING WITH BERT” by Rodrigo Nogueira et al. [7] technique for re-ranking passages in information retrieval systems using BERT (Bidirectional Encoder Representations from Transformers) models is presented in the work.

The relevance of re-ranking passages to increase the precision of information retrieval systems is emphasized in the paper, which also covers the difficulties in creating efficient re-ranking models. In the suggested method, the query and candidate passages are encoded using a BERT model, and a relevance score is then calculated between the query and each passage. The texts are then re-examined in terms of their relevance to the query using the relevance scores. The authors assess their method using two common datasets and contrast the outcomes with a number of benchmarks. The results of the evaluation show that the proposed approach significantly improves accuracy relative to baseline models and surpasses state-of-the-art passage reclassification methods. The paper concludes that BERT-based passage re-ranking can be an effective technique for improving the accuracy of information retrieval systems and has potential applications in various domains such as search engines, question answering systems, and chatbots.

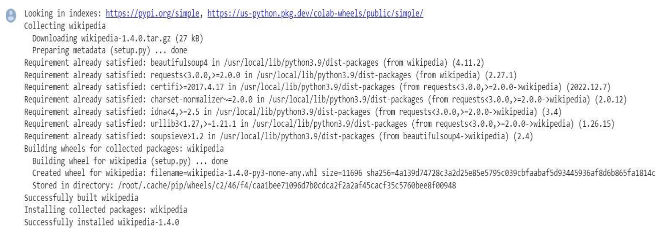
Overall, this paper makes an important contribution to information research by adapting BERT as a re-ranker passage. BERT-based models surpass the previous state-of-the-art models by a large margin. This will help the user to give them relevant documents according to their query.

III. Novelty

To improve the search results, users can give the feedback for each search result whether it is relevant or not. Based on the user feedback, refine the search engine results continuously and enhance the performance over time.

IV. Methodology:

4.1 Data Collection: The data is collected using the Wikipedia API. The search query and language are given as input to the API, and it returns the search results as JSON data.

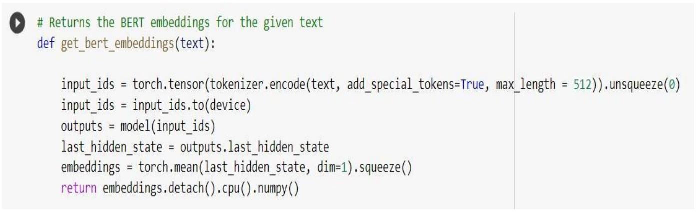


```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting wikipedia
  Downloading wikipedia-1.4.0.tar.gz (27 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.9/dist-packages (from wikipedia) (4.11.2)
Requirement already satisfied: requests<3.0.0,>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from wikipedia) (2.27.1)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.0.0->wikipedia) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.0.0->wikipedia) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.0.0->wikipedia) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.0.0->wikipedia) (1.26.15)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.9/dist-packages (from beautifulsoup4->wikipedia) (2.4)
Building wheels for collected packages: wikipedia
  Building wheel for wikipedia (setup.py) ... done
  Created wheel for wikipedia: filename=wikipedia-1.4.0-py3-none-any.whl size=11696 sha256=4a139d74728c3a2d25e85e5795c039cbfaabaf5d93445936af8d6b865fa1814c
  Stored in directory: /root/.cache/pip/wheels/c2/46/f4/caa1bee71096d7b0cdca2f2a2af45cacf35c5760bee8f00948
Successfully built wikipedia
Installing collected packages: wikipedia
Successfully installed wikipedia-1.4.0
```

Fig 1: Data Collection

4.2 Data Preprocessing: The text data obtained from the API is preprocessed by removing the HTML tags, punctuations, stop words, and numbers. The text data is then tokenized into words.In general, for BERT-based models, the data preprocessing step typically involves tokenization of the input text], and converting the text into numerical input embeddings that can be fed into the BERT model. This can be done using the tokenizer provided by the transformers library in Python.

4.3 Embedding Generation: The preprocessed text data is fed into the BERT model to generate contextualized word embeddings. These embeddings capture the contextual meaning of the words in the documents, which is useful for measuring their similarity.

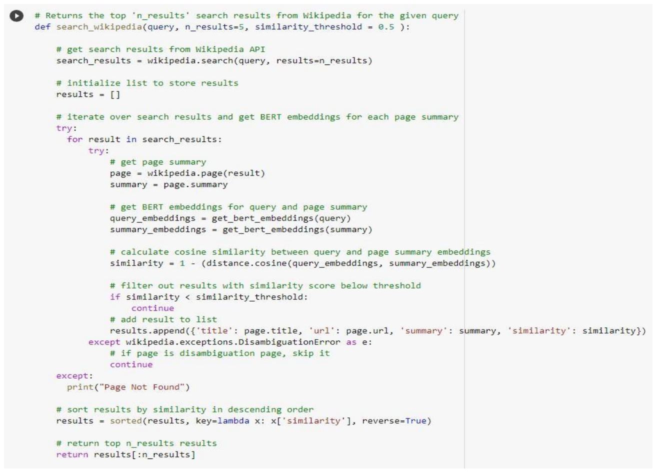


```python
# Returns the BERT embeddings for the given text
def get_bert_embeddings(text):

    input_ids = torch.tensor(tokenizer.encode(text, add_special_tokens=True, max_length = 512)).unsqueeze(0)
    input_ids = input_ids.to(device)
    outputs = model(input_ids)
    last_hidden_state = outputs.last_hidden_state
    embeddings = torch.mean(last_hidden_state, dim=1).squeeze()
    return embeddings.detach().cpu().numpy()
```

Fig 2: Embedding Generation

4.4 Search using the Embeddings: The search query is also fed into the BERT model to generate an embedding for it. The cosine similarity is calculated between the query embedding and the embeddings of each document. The documents with the highest cosine similarity scores are considered the most relevant and are returned as search results.



```python
# Returns the top 'n_results' search results from Wikipedia for the given query
def search_wikipedia(query, n_results=5, similarity_threshold = 0.5 ):

    # get search results from Wikipedia API
    search_results = wikipedia.search(query, results=n_results)

    # initialize list to store results
    results = []

    # iterate over search results and get BERT embeddings for each page summary
    try:
      for result in search_results:
          try:
              # get page summary
              page = wikipedia.page(result)
              summary = page.summary

              # get BERT embeddings for query and page summary
              query_embeddings = get_bert_embeddings(query)
              summary_embeddings = get_bert_embeddings(summary)

              # calculate cosine similarity between query and page summary embeddings
              similarity = 1 - (distance.cosine(query_embeddings, summary_embeddings))

              # filter out results with similarity score below threshold
              if similarity < similarity_threshold:
                  continue
              # add result to list
              results.append({'title': page.title, 'url': page.url, 'summary': summary, 'similarity': similarity})
          except wikipedia.exceptions.DisambiguationError as e:
              # if page is disambiguation page, skip it
              continue
    except:
      print("Page Not Found")

    # sort results by similarity in descending order
    results = sorted(results, key=lambda x: x['similarity'], reverse=True)

    # return top n_results results
    return results[:n_results]
```

Fig 3: Search using the Embedding

V. Evaluation

To evaluate the performance of the search engine, we calculate the Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) for the top k search results. MAP measures the average precision of the search engine across multiple queries, while MRR measures the rank of the first relevant result.

Precision: It is a measure of fraction of retrieved docs that are relevant = P(relevant|retrieved).

Recall: It is the fraction of relevant docs that are retrieved = P(retrieved|relevant).

f1-Score: It is the harmonic mean of a system's precision and recall values.

Accuracy: It is expressed as a percentage of the number of relevant documents retrieved out of the total number of documents retrieved.

AP (Average precision): It is the mean of the precision scores after each relevant document is retrieved.

mAP (mean average precision): It is the average of AP.

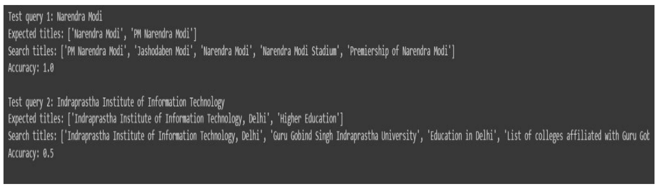
MRR (Mean Reciprocal Rank): It is superior to other evaluation metrics because it takes into consideration the position of the relevant document in the list of documents that were retrieved, which is crucial to the user's perception of the system. MRR additionally tracks how quickly a user locates relevant information, whereas other evaluation metrics do not.

Query-1: Narendra Modi

Evaluation Metric	**TF-IDF**	**BERT**
Precision	0.20	0.4
Recall	1.0	1.0
F1-Score	0.33	0.5714
Accuracy	0.0	1.0
MAP	0.0	0.83
MAR	0.0	1.0

Query-2: Indraprastha Institute of Technology

Evaluation Metric	**TF-IDF**	**BERT**
Precision	0.10	0.2
Recall	0.50	0.5
F1-Score	0.17	0.2857
Accuracy	0.0	0.5
MAP	0.0	0.6666
MAR	0.0	1.0



```
Test query 1: Narendra Modi
Expected titles: ['Narendra Modi', 'PM Narendra Modi']
Search titles: ['PM Narendra Modi', 'Jashodaben Modi', 'Narendra Modi', 'Narendra Modi Stadium', 'Premiership of Narendra Modi']
Accuracy: 1.0

Test query 2: Indraprastha Institute of Information Technology
Expected titles: ['Indraprastha Institute of Information Technology, Delhi', 'Higher Education']
Search titles: ['Indraprastha Institute of Information Technology, Delhi', 'Guru Gobind Singh Indraprastha University', 'Education in Delhi', 'List of colleges affiliated with Guru Gob
Accuracy: 0.5
```

Fig 4: Search Result

The first perception of a user is the title of the search result so the evaluation is to be done on the titles generated by the model.So we can observe in the above image that the titles generated in other embedding techniques are irrelevant while the titles generated using BERT are comparatively relevant.

VI. Result

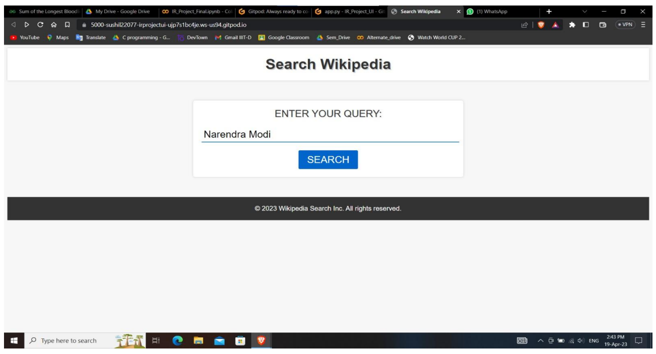
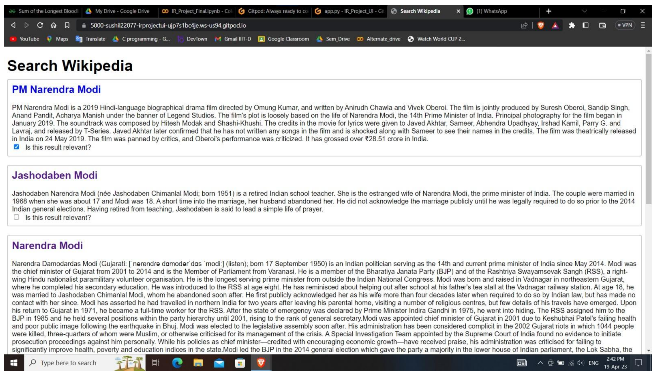


Fig 5: Home page

As shown in Fig 5 it is the home page in which we just need to write the query and click on search to get the results for the query.



Search Wikipedia

PM Narendra Modi

PM Narendra Modi is a 2019 Hindi-language biographical drama film directed by Omung Kumar, and written by Anirudh Chawla and Vivek Oberoi. The film is jointly produced by Suresh Oberoi, Sandip Singh, Anand Pandit, Acharya Manish under the banner of Legend Studios. The film's plot is loosely based on the life of Narendra Modi, the 14th Prime Minister of India. Principal photography for the film began in January 2019. The soundtrack was composed by Hitesh Modak and Shashi-Khushi. The credits in the movie for lyrics were given to Javed Akhtar, Sameer, Abhendra Upadhyay, Irshad Kamil, Parry G. and Lavraj, and released by T-Series. Javed Akhtar later confirmed that he has not written any songs in the film and is shocked along with Sameer to see their names in the credits. The film was theatrically released in India on 24 May 2019. The film was panned by critics, and Oberoi's performance was criticized. It has grossed over ₹28.51 crore in India.

☑ Is this result relevant?

Jashodaben Modi

Jashodaben Narendra Modi (née Jashodaben Chimanlal Modi; born 1951) is a retired Indian school teacher. She is the estranged wife of Narendra Modi, the prime minister of India. The couple were married in 1968 when she was about 17 and Modi was 18. A short time into the marriage, her husband abandoned her. He did not acknowledge the marriage publicly until he was legally required to do so prior to the 2014 Indian general elections. Having retired from teaching, Jashodaben is said to lead a simple life of prayer.

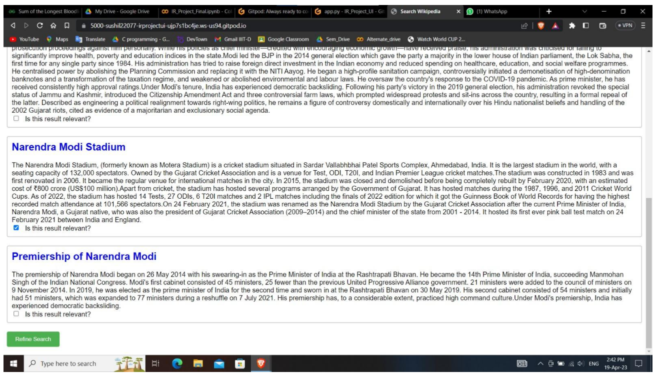
☐ Is this result relevant?

Narendra Modi

Narendra Damodardas Modi (Gujarati: [ˈnəɾendɾə dɑmodəɾˈdɑs ˈmodiː] (listen); born 17 September 1950) is an Indian politician serving as the 14th and current prime minister of India since May 2014. Modi was the chief minister of Gujarat from 2001 to 2014 and is the Member of Parliament from Varanasi. He is a member of the Bharatiya Janata Party (BJP) and of the Rashtriya Swayamsevak Sangh (RSS), a right-wing Hindu nationalist paramilitary volunteer organisation. He is the longest serving prime minister from outside the Indian National Congress. Modi was born and raised in Vadnagar in northeastern Gujarat, where he completed his secondary education. He was introduced to the RSS at age eight. He has reminisced about helping out after school at his father's tea stall at the Vadnagar railway station. At age 18, he was married to Jashodaben Chimanlal Modi, whom he abandoned soon after. He first publicly acknowledged her as his wife more than four decades later when required to do so by Indian law, but has made no contact with her since. Modi has asserted he had travelled in northern India for two years after leaving his parental home, visiting a number of religious centres, but few details of his travels have emerged. Upon his return to Gujarat in 1971, he became a full-time worker for the RSS. After the state of emergency was declared by Prime Minister Indira Gandhi in 1975, he went into hiding. The RSS assigned him to the BJP in 1985 and he held several positions within the party hierarchy until 2001, rising to the rank of general secretary.Modi was appointed chief minister of Gujarat in 2001 due to Keshubhai Patel's failing health and poor public image following the earthquake in Bhuj. Modi was elected to the legislative assembly soon after. His administration has been considered complicit in the 2002 Gujarat riots in which 1044 people were killed, three-quarters of whom were Muslim, or otherwise criticised for its management of the crisis. A Special Investigation Team appointed by the Supreme Court of India found no evidence to initiate prosecution proceedings against him personally. While his policies as chief minister—credited with encouraging economic growth—have received praise, his administration was criticised for failing to significantly improve health, poverty and education indices in the state.Modi led the BJP in the 2014 general election which gave the party a majority in the lower house of Indian parliament, the Lok Sabha, the

Fig 6: Search results

As shown in Fig 6 the results of the entered query by user are shown in the list form and also a checkbox is given for each result which is used as user feedback.



significantly improve health, poverty and education indices in the state.Modi led the BJP in the 2014 general election which gave the party a majority in the lower house of Indian parliament, the Lok Sabha, the first time for any single party since 1984. His administration has tried to raise foreign direct investment in the Indian economy and reduced spending on healthcare, education, and social welfare programmes. He centralised power by abolishing the Planning Commission and replacing it with the NITI Aayog. He began a high-profile sanitation campaign, controversially initiated a demonetisation of high-denomination banknotes and a transformation of the taxation regime, and weakened or abolished environmental and labour laws. He oversaw the country's response to the COVID-19 pandemic. As prime minister, he has received consistently high approval ratings.Under Modi's tenure, India has experienced democratic backsliding. Following his party's victory in the 2019 general election, his administration revoked the special status of Jammu and Kashmir, introduced the Citizenship Amendment Act and three controversial farm laws, which prompted widespread protests and sit-ins across the country, resulting in a formal repeal of the latter. Described as engineering a political realignment towards right-wing politics, he remains a figure of controversy domestically and internationally over his Hindu nationalist beliefs and handling of the 2002 Gujarat riots, cited as evidence of a majoritarian and exclusionary social agenda.

☐ Is this result relevant?

Narendra Modi Stadium

The Narendra Modi Stadium, (formerly known as Motera Stadium) is a cricket stadium situated in Sardar Vallabhbhai Patel Sports Complex, Ahmedabad, India. It is the largest stadium in the world, with a seating capacity of 132,000 spectators. Owned by the Gujarat Cricket Association and is a venue for Test, ODI, T20I, and Indian Premier League cricket matches.The stadium was constructed in 1983 and was first renovated in 2006. It became the regular venue for international matches in the city. In 2015, the stadium was closed and demolished before being completely rebuilt by February 2020, with an estimated cost of ₹800 crore (US$100 million).Apart from cricket, the stadium has hosted several programs arranged by the Government of Gujarat. It has hosted matches during the 1987, 1996, and 2011 Cricket World Cups. As of 2022, the stadium has hosted 14 Tests, 27 ODIs, 6 T20I matches and 2 IPL matches including the finals of 2022 edition for which it got the Guinness Book of World Records for having the highest recorded match attendance at 101,566 spectators.On 24 February 2021, the stadium was renamed as the Narendra Modi Stadium by the Gujarat Cricket Association after the current Prime Minister of India, Narendra Modi, a Gujarat native, who was also the president of Gujarat Cricket Association (2009–2014) and the chief minister of the state from 2001 - 2014. It hosted its first ever pink ball test match on 24 February 2021 between India and England.

✓ Is this result relevant?

Premiership of Narendra Modi

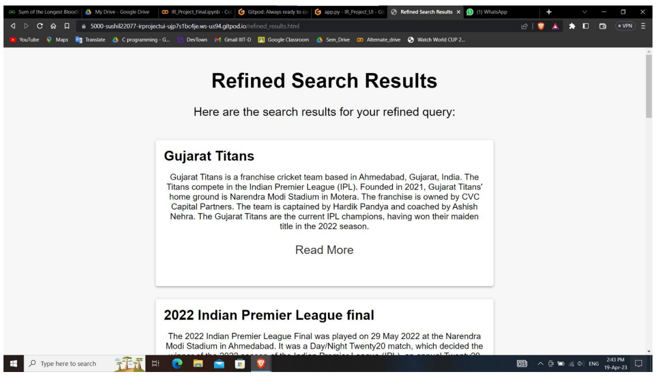
The premiership of Narendra Modi began on 26 May 2014 with his swearing-in as the Prime Minister of India at the Rashtrapati Bhavan. He became the 14th Prime Minister of India, succeeding Manmohan Singh of the Indian National Congress. Modi's first cabinet consisted of 45 ministers, 25 fewer than the previous United Progressive Alliance government. 21 ministers were added to the council of ministers on 9 November 2014. In 2019, he was elected as the prime minister of India for the second time and sworn in at the Rashtrapati Bhavan on 30 May 2019. His second cabinet consisted of 54 ministers and initially had 51 ministers, which was expanded to 77 ministers during a reshuffle on 7 July 2021. His premiership has, to a considerable extent, practiced high command culture.Under Modi's premiership, India has experienced democratic backsliding.

☐ Is this result relevant?

Refine Search

Fig 7: Search results

As shown in Fig 7 there is a button for refined search which will process the results according to the user feedback.


Refined Search Results
5000-sushil22077-irprojectui-ujp7s1bc4je.ws-us94.gitpod.io/refined_results.html
Type here to search
2:43 PM
19-Apr-23

Refined Search Results

Here are the search results for your refined query:

Gujarat Titans

Gujarat Titans is a franchise cricket team based in Ahmedabad, Gujarat, India. The Titans compete in the Indian Premier League (IPL). Founded in 2021, Gujarat Titans' home ground is Narendra Modi Stadium in Motera. The franchise is owned by CVC Capital Partners. The team is captained by Hardik Pandya and coached by Ashish Nehra. The Gujarat Titans are the current IPL champions, having won their maiden title in the 2022 season.

Read More

2022 Indian Premier League final

The 2022 Indian Premier League Final was played on 29 May 2022 at the Narendra Modi Stadium in Ahmedabad. It was a Day/Night Twenty20 match, which decided the

Fig 8: Refined Search results

As shown in Fig 7 the refined search results are shown.

References

[1] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, “Using of Jaccard Coefficient for Keywords Similarity,” Hong Kong, 2013.

[2] A. Khatri and P. P, “Sarcasm Detection in Tweets with BERT and GloVe Embeddings,” in Proceedings of the Second Workshop on Figurative Language Processing, Online: Association for Computational Linguistics, 2020, pp. 56–60. doi: 10.18653/v1/2020.figlang-1.7.

[3] R. Nogueira and K. Cho, “Passage Re-ranking with BERT.” arXiv, Apr. 14, 2020. Accessed: Apr. 19, 2023. [Online]. Available: http://arxiv.org/abs/1901.04085

[4] M. Hu, E.-P. Lim, A. Sun, H. W. Lauw, and B.-Q. Vuong, “On Improving Wikipedia Search using Article Quality”.

[5] M. Kanakaraj and S. S. Kamath, “NLP based intelligent news search engine using information extraction from e-newspapers,” in 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India: IEEE, Dec. 2014, pp. 1–5. doi: 10.1109/ICCIC.2014.7238500.

[6] D. Miller, “Leveraging BERT for Extractive Text Summarization on Lectures”.

[7] D. R. Kubal and A. V. Nimkar, “A survey on word embedding techniques and semantic similarity for paraphrase identification,” IJCSYSE, vol. 5, no. 1, p. 36, 2019, doi: 10.1504/IJCSYSE.2019.098417.